

Desarrollo de un prototipo de videojuego de lucha utilizando la realidad aumentada

Cristian Fernández Suárez

Resumen—En la actualidad existen multitud de videojuegos para móviles debido al boom de la telefonía móvil y a las facilidades que existen para poder crear tu propio videojuego y colgarlo en la tienda oficial (Google Play o app store). Pero en el ámbito de la realidad aumentada hay pocos videojuegos para móviles, este tipo de videojuegos suelen ser más numerosos en consolas portátiles e incluso en consolas de sobremesa, debido especialmente a la potencia del hardware.

Para este proyecto se ha decidido llevar la realidad aumentada a los móviles en forma de videojuego de lucha con elementos *Role Playing Game* (RPG).

Para desarrollar este proyecto se han utilizado las librerías de Vuforia, el programa de modelado Blender, Unity para darle la lógica y animar a los personajes y Gimp para hacer los fondos de pantalla y botones.

Puesto que el tiempo de desarrollo es muy corto solo se llegará a realizar un prototipo, con el cual en un futuro se puede avanzar añadiéndole funcionalidades y material nuevo (como personajes o ataques nuevos).

Palabras clave— Android, aplicación móvil, AR, Unity, Blender, Gimp, Vuforia, combate por turnos, RPG, C#, XML

Abstract—At present there are many mobile games due to the boom in mobile telephony and the facilities available to create your own game and hang it on the official store (Google Play or App Store). But in the realm of augmented reality there are few mobile games, this type of games are often more numerous in portables consoles and even on home consoles, especially due to the power of the hardware.

For this project it was decided to bring augmented reality to mobile as a fighting game with Role Playing Game (RPG) features.

To develop this project have been used Vuforia libraries, the program of modelling Blender, Unity to give logic and animation to characters and Gimp for the design of buttons and background images.

As the development time is very short, we can only make a prototype, with which in the future can be advanced by adding features and new material (such as characters or new attacks).

Index Terms— Android, mobile application, AR, Unity, Blender, Gimp, Vuforia, RPG, turn-based combat, C#, XML



1 INTRODUCCIÓN

EN la actualidad, la industria de los videojuegos ha evolucionado enormemente. Ahora el realismo y las mecánicas de juego, en la mayoría de casos, nada tienen que ver con los primeros videojuegos, los cuales eran en 2 dimensiones y los controles eran muy simples.

Esto es debido a que la tecnología avanza y somos capaces de poder trabajar con máquinas más potentes dando al usuario más realismo y libertad.

El *gameplay* (o mecánica de juego), está en continua evolución, añadiendo nuevas formas de jugar o utilizando hardware específico para hacer que el usuario sienta más cercana la jugabilidad (como por ejemplo la interacción con Kinect o la tecnología que está empezando a despuntar en estos momentos, la realidad virtual).

El mayor realismo logrado hasta la fecha se ha conseguido mediante gráficos más potentes y con la utilización de *shaders*, junto a una iluminación apropiada.

Desde el salto de las 2 dimensiones a las 3 dimensiones se ha ido perfeccionando este realismo pero actualmente estamos en un punto de estancamiento. El realismo no puede mejorar mucho más desde el apartado gráfico, con lo cual se opta por utilizar otras tecnologías que hagan de la experiencia del usuario algo más realista si cabe. Actualmente estas tecnologías vendrían encabezada por la realidad virtual (VR), con los proyectos de Oculus Rift[1] y Project Morpheus[2] a la cabeza, y con la captación de

movimientos, con Kinect como herramienta más popular. Sin embargo, la tecnología de realidad aumentada (AR) no ha sido apenas explotada para darle más realismo o un *gameplay* innovador a un videojuego.

Este Trabajo Final de Grado (TFG) pretende utilizar la tecnología de AR para desarrollar un prototipo de videojuego de lucha con elementos de *Role Playing Game* (RPG). De esta forma se quiere explorar y ahondar en el mundo de la AR aplicándolo a un ámbito en el que se ha extendido poco esta tecnología, cómo es el de los videojuegos.

¿QUÉ ES LA REALIDAD AUMENTADA (AR)?

La realidad aumentada (AR) es la forma de potenciar los sentidos del ser humano añadiéndole imágenes o información generada virtualmente sobre el mundo real, complementando así el mundo real con el digital. Debido a esta fusión de mundos, el usuario puede llegar a olvidarse de que ha sido creado digitalmente y que forma parte de la realidad.[3]

La tecnología de AR ha tenido especial incidencia en ámbitos tan dispares como la educación, la cultura, la medicina o el entretenimiento:

En educación, por ejemplo en museos o exhibiciones se utiliza esta tecnología aprovechando las conexiones Wireless para mostrar información sobre objetos o lugares, así como imágenes virtuales como por ejemplo ruinas reconstruidas.

“Existen también interesantes proyectos educativos, como el desarrollado por la Unidad LabHuman [4], del Instituto I3BH, en la Universidad Politécnica de Valencia. Sus investigadores han creado unos cuadernos que incluyen unas cartulinas con un código. Estas sirven para que una webcam coloque en el monitor donde trabaja el alumno una imagen 3D acorde con los contenidos de la lección que está estudiando. La imagen se puede girar, acercar y alejar, ver desde distintas perspectivas”

En medicina, “los sistemas de Realidad Aumentada pueden facilitar el trabajo en campos como la cirugía. A través de resonancias magnéticas es posible recabar datos del interior del paciente de una manera no invasiva, además de realizar una reconstrucción que puede ser superpuesta sobre el cuerpo físico en tiempo real. De esta manera, se pueden conseguir operaciones más rápidas y con mayores garantías de seguridad para los pacientes.”[5]

En entretenimiento, se utiliza sobretodo en eventos deportivos, poniendo publicidad en el centro del campo de fútbol o poniendo la barra del récord vigente en una carrera de atletismo.

Sin embargo, en el ámbito de los videojuegos esta tecnología se ha utilizado poco.

1.1 Mercado actual de videojuegos con AR

Como ya se ha comentado antes, la AR es una tecnología relativamente joven que cuenta con varios proyectos interesantes, pero que en el mundo de los videojuegos quizás ha incidido menos, siendo utilizada para proyectos más innovadores que convencionales (no se ha utilizado para sagas de videojuegos contrastadas, sino más bien para videojuegos pioneros).

Tenemos el ejemplo de los siguientes videojuegos:

El caso de Invizimals[6] es un videojuego para PSP creado por la compañía Novarama y comercializado por Sony en que se utiliza la innovación de la realidad aumentada, el juego incluye una plataforma en forma de tarjeta sobre la que aparecerán los “invizimals” y una cámara que conecta con el puerto frontal de la PSP.

El juego se basa en ir capturando monstruos y después poder utilizarlos en combate, con un claro estilo “Pokémon”. Además tiene una pequeña trama para darle un modo historia.

Wonderbook[7] es un proyecto que llevó a cabo Sony junto a la autora de Harry Potter, J.K. Rowling, para hacer un libro de hechizos con la ayuda de la AR.

Este videojuego necesita de la cámara PlayStation Eye y del mando PS Move, este último servirá a modo de varita. Con la AR crean los efectos de los hechizos en tu propia

realidad. Además el libro de hechizos es en cada página una tarjeta codificada, donde se recrean modelos 3D.

EyePet[8] es un juego exclusivo de PlayStation 3 y de PlayStation Portable, desarrollado por London Studio y Playlogic Game Factory.

Este juego usa la realidad aumentada con la PlayStation Eye (cámara de PlayStation) para mostrarnos una mascota virtual con la que podremos interactuar con las manos, con una tarjeta de AR incluida en el juego o con el mando PlayStation Move.

Por último, Droid Shooting es un videojuego de realidad aumentada para móviles desarrollado por Quest-Com. Es un videojuego de tipo “arcade” en el que el objetivo es disparar a unos androides (que son el logo de Android) que irán apareciendo a tu alrededor. Este juego a diferencia de los anteriores no utiliza tarjetas de AR.

El proyecto que se propone en este TFG es un juego de lucha similar a Invizimals, con marcadores que representen cada personaje, pero con unas mecánicas de juego distintas, basándose nuestro proyecto en el combate por turnos y con algunos aspectos RPG que Invizimals no posee.

1.2 Motivación de la utilización del móvil

Para este proyecto de AR se necesita una cámara móvil, ya que se debe enfocar a las tarjetas codificadas para que los modelos 3D aparezcan. Una cámara estática como la de un ordenador no serviría para enfocar estas tarjetas o el mundo real en general, solo enfoca un punto o tiene poca libertad para enfocar diferentes sitios. Entonces se ha decidido que lo ideal es utilizar un dispositivo portátil (dispositivo móvil o consola portátil).

Analizando las ventajas y desventajas de cada dispositivo tenemos lo siguiente.

Ventajas y desventajas de utilizar un dispositivo móvil:

Las ventajas son que casi todo el mundo tiene un móvil, así que el mercado de posibles usuarios es muy amplio, y que el desarrollo es gratuito (dependiendo del entorno de desarrollo que se escoja puede no serlo).

La desventaja principal de los dispositivos móviles es que prácticamente todos tienen especificaciones más bajas respecto a consolas portátiles.

Ventajas y desventajas de utilizar una consola portátil:

Las ventajas principales son que normalmente tienen mejores especificaciones que los móviles y que está mejor preparada, tanto la pantalla como la cámara, para dar una buena jugabilidad.

Las desventajas principales son por un lado, que hay mucho menos mercado, ya que respecto a los móviles, mucha menos gente tiene una consola portátil en concreto. Además, algunas consolas portátiles no tienen cámara

integrada, se compra por separado, lo que acotaría más el mercado. Por otro lado el desarrollo para una consola no es gratuito debido a las herramientas que se tienen que utilizar.

Analizando estas ventajas y desventajas se ha elegido utilizar el móvil, ya que su desarrollo es gratis (en Android) y el producto puede llegar a más gente.

-
- E-mail de contacte: cristian.fernandezsua@gmail.com
 - Menció realitzada: Enginyeria del Software
 - Treball tutoritzat per: Dèbora Gil i Antoni Gurgui (departament de computació)
 - Curs 2014/15

2 OBJETIVOS

El objetivo general que se propuso para el proyecto era el de lograr hacer un prototipo de videojuego de realidad aumentada, del género lucha con elementos RPG y para dispositivos móviles con sistema operativo Android.

El desarrollo del videojuego se hará cubriendo 3 sub-objetivos principales:

Objetivo 1: análisis de los requisitos:

Para saber las expectativas que los usuarios finales pueden llegar a tener sobre el proyecto, se realizó una encuesta a los usuarios potenciales.

Seguidamente se categorizaron los requisitos según si son funcionales/no funcionales y según el modelo de Kano.

Objetivo 2: Diseño e implementación de las funcionalidades básicas del videojuego.

El juego consta de 4 módulos de desarrollo:

1- Módulo AR:

En este módulo hay que desarrollar un entorno AR donde el usuario pueda interactuar con los personajes mediante botones. Los personajes serán representados gracias a unas tarjetas codificadas que deberemos decidir analizando las características de las tarjetas.

2- Módulo Visualización

En este módulo se tienen que diseñar y modelar a los personajes y posteriormente crear sus animaciones y las de los ataques.

3- Módulo lógica del combate

Hay que implementar un combate por turnos en el que se deban realizar ataques hasta que uno de los personajes se quede sin vida.

Para guardar los datos tendremos que elegir una forma de persistencia de datos y desarrollar la misma.

Se elaboraran diagramas de flujo y de

clases antes de empezar la codificación.

4- Módulo de GUI

Se creará la GUI de los modos, analizando previamente como se debe organizar cada pantalla.

Objetivo 3: análisis de recursos:

Se deberá realizar un buen cálculo de recursos necesarios (especificaciones del dispositivo) para utilizar el producto final con un rendimiento óptimo. Así conseguiremos un producto fiable en los dispositivos deseados.

3 REQUISITOS DE LA APLICACIÓN

Al inicio del proyecto se analizaron los requisitos que los usuarios potenciales demandaban para poder enfocar el proyecto lo máximo posible a sus gustos.

3.1 Modo de captación de requisitos

Se activó una encuesta para saber cuáles eran los gustos de los usuarios, y así poder enfocar el proyecto a sus gustos, todo y manteniendo unas restricciones. La encuesta se hizo mediante "SurveyMonkey". En el siguiente enlace se puede encontrar la encuesta en cuestión: <https://es.surveymonkey.com/r/R3BPVFZ>.

3.2 Resultados de la encuesta

La encuesta nos arrojó los siguientes datos sobre los usuarios potenciales:

- **Edad de los usuarios:** entre 17 y 25 años.
- **Plataforma:** Según las restricciones estaba pensado para Android, pero los usuarios lo reafirmaron con un 84%.
- **Tiempo de juego:** Menos de 1 hora.
- **Lugar de juego:** Casa y transportes.
- **Género preferidos RPG,** aventura y plataformas
- **¿Qué valora el usuario en un videojuego?:** Diversión y fácil manejo.
- **Redes sociales:** Prefieren no compartir nada del videojuego en las redes sociales.

3.3 Análisis de los requisitos captados

Una vez tenemos las opiniones de los usuarios y los requisitos restrictivos que se pusieron antes de comenzar el proyecto se ha hecho una categorización de estos requisitos mediante un modelo de Kano y separándolos entre funcionales y no funcionales.

3.4 Categorización de requisitos

Modelo de Kano de los requisitos:

Dissatisfiers:

El videojuego tiene que funcionar en un dispositivo móvil con SO Android.

Las partidas deben poder ser rápidas, ya que según los datos de las encuestas no juegan durante mucho tiempo.

La interfaz gráfica tiene que ser intuitiva, con solo ver los iconos de los botones deben saber qué hace ese botón.

Satisfiers:

Tener implementado el modo museo.

Tener implementado el modo combate contra CPU.

Tener una buena documentación que respalde el proyecto.

Hacer unos test lo más completos posible y de diferentes tipo (utilizar tanto "beta testers" como la técnica de "test driven development") para lograr un software fiable.

Delighters:

Implementar los modos multijugador.

Implementar un modo historia (modo 1 jugador).

Aplicar alguna funcionalidad extra más como podría ser poner estadísticas.

Lograr unos gráficos lo más refinados posible y que no lacre el "gameplay".

Requisitos funcionales - no funcionales:

Requisitos funcionales:

El usuario puede visionar sus cartas en el modo museo (de una en una).

El usuario puede hacer fotos en el modo museo, que se guardarán en el propio dispositivo.

El usuario puede realizar combates contra la máquina, utilizando sus cartas.

El usuario NO podrá realizar ataques con tarjetas de ataque que no pueda utilizar el personaje en cuestión.

Los personajes subirán de nivel según ganan batallas.

Cada usuario tendrá sus personajes guardados en una BD local en su dispositivo, en el que incluirá el nivel del personaje y sus atributos.

El sistema proporcionará opciones para audio y gráficos.

Requisitos NO funcionales:

El tiempo de respuesta tiene que ser muy bajo (que no haya lag, ya que repercutiría mucho el gameplay).

El videojuego funcionará en sistemas Android.

No debe tener un gran consumo de batería (nos ceñiremos al número de horas que los usuarios han dicho que juegan en la encuesta).

Las actualizaciones deben ser pensadas a fondo, no puede ser que el juego no esté disponible muchas veces porque necesite actualizarse.

Para un usuario cualquiera debe serle fácil de aprender a utilizar la interfaz en unos minutos.

4 DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN

La metodología de desarrollo a utilizar en el proyecto es SCRUM.

Scrum es una metodología de desarrollo ágil utilizada por equipos de desarrollo para aumentar la competitividad, eficiencia, productividad y flexibilidad en las tareas. Se basa en "sprints" de 1 mes o 2 semanas y en una reunión diaria con el equipo para ver el avance de las tareas.

4.1 Diagrama de módulos

Para el desarrollo del proyecto se ha hecho el siguiente diagrama de módulos en el cual se dividen las tareas llevadas a cabo según su ámbito.

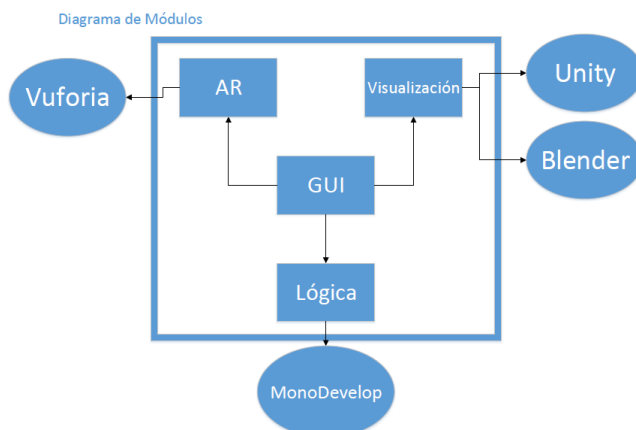


Figura 1: Diagrama de módulos

Como se puede ver en la figura 1, la arquitectura de desarrollo cuenta con 4 módulos (son los rectángulos) y cada uno de estos módulos cuenta con una o más de una herramientas externas con las que se implementa (son las elipses).

- Módulo de GUI: en este módulo se diseñan los botones y títulos, para posteriormente implementarlos en el programa mediante código C#.

Para el diseño tanto de botones como de títulos se ha utilizado el software libre y gratuito de edición de imágenes Gimp.

- Módulo de AR: las tareas de este módulo van ligadas a la librería Vuforia[9], librería que nos proporciona la AR. En este módulo se han realizado las tareas para pasar un modelo 3D a la AR y se han tomado las decisiones de qué códigos se debían poner en las tarjetas codificadas que representan el modelo 3D.

- Módulo de visualización: en este módulo se modelan los personajes y sus animaciones.

Para el modelado del personaje (malla) se ha utilizado Blender[10], ya que es un software de modelado gráfico muy potente, que permite exportar las mallas a la mayoría de formatos.

Para la animación de los personajes se ha utilizado Unity[11], software de referencia en animación y videojuegos que permite la implementación del *gameplay* mediante el lenguaje de programación C# o Javascript.

- Módulo de lógica: en este módulo se contempla básicamente la parte del control del combate. Esta parte se ha desarrollado con el compilador MonoDevelop, que ya viene integrado con Unity, y se ha utilizado como lenguaje de programación C#.

4.2 Módulo de Visualización

En este módulo se diseñan los personajes del videojuego y sus animaciones.

Para el diseño de los personajes se ha utilizado Blender, mientras que las animaciones se han implementado en Unity.

En este primer prototipo solo disponemos de 2 personajes: Froyo (el personaje que controlamos) y el monstruo (al que debemos vencer con Froyo).

Entonces, una de las tareas de este módulo es el diseño de los monstruos. El monstruo que he diseñado es un monstruo volador, lo he hecho así por simpleza, así me evitaba el movimiento de piernas y el número de animaciones era menor y más simple, todo esto repercute en el consumo de recursos del móvil.

El monstruo enemigo se descargó de la tienda de Unity[12] de forma gratuita, ya que la importancia de mi TFG no reside en el diseño sino en el trabajo de ingeniería.

La otra tarea principal de este módulo es la creación de las animaciones de los personajes. Las animaciones han sido creadas con la herramienta "Animation" de Unity. Estas animaciones han sido creadas para el personaje diseñado, para el que se descargó a través de Unity no, ya que el personaje venía con las animaciones creadas.

Las animaciones son clips de formato ".anim" que son llamados desde el código. En el caso de la animación de movimiento continuo, es un clip ejecutado en modo "loop", se llama solo una vez y el clip se ejecutará mientras no se llame a otro clip del mismo personaje. El resto de animaciones solo se ejecuta una vez (cuando el usuario toca el botón), una vez finalizada la animación no vuelve a repetirla si no tocamos el mismo botón.

Las animaciones creadas son cortas para dar más dinamismo al combate y que el usuario no tenga que estar sin hacer nada durante mucho tiempo.

4.3 Módulo de AR

En este módulo se encuentran las tareas para pasar un modelo 3D a la realidad aumentada. Para lograrlo se ha utilizado la librería Vuforia de Qualcomm junto a los códigos QR.

Se ha decidido utilizar como marcadores los códigos QR. Estos códigos son muy utilizados en la actualidad para mostrar información a los usuarios por parte de empresas, para hacer el recuento de inventario en tiendas y para los videojuegos de AR.[13]

Se han decidido utilizar estos marcadores porque tienen un buen balance de "tiempo de detección"-robustez de la imagen.

Los códigos QR se han elegido según el *rating* de la sección "target manager" de Vuforia, donde se cuelgan las imágenes a utilizar como tarjetas de AR y la propia Web hace un *rating* de cómo de bien va esa imagen.

Para hacer esto, valora la imagen entre 0 y 5 estrellas. Esta valoración la hace mediante un algoritmo interno que no se conoce, pero la propia Vuforia da las características de la imagen que se contemplan en el algoritmo, son las siguientes:

- **Características de la imagen:** Los detalles agudos como pinchos son las características de la imagen. Hay que utilizar imágenes que tengan muchas características, así logramos una mayor robustez (que la imagen no desaparezca si tapamos parte del código QR).
- **Contraste:** Cuanto mayor contraste mejor detección.
- **Distribución de las características:** Las características tienen que estar distribuidas por toda la imagen. En otro caso si tapásemos la parte del código con las características, el modelo desaparecería.
- **Evitar formas suaves o redondeadas.**
- **Evitar repetición de patrones.**

En la figura 2 se pueden ver los códigos QR utilizados.



Figura 2: Tarjetas AR utilizadas

4.4 Módulo de GUI

Las tareas de este módulo están relacionadas con la interfaz de usuario y con el análisis de recursos, ya que este último tendría contacto con todos los módulos al igual que el módulo de GUI se ha decidido juntarlos.

Para el diseño de botones y títulos se ha utilizado Gimp y para su implementación C#.

Para evitar problemas de derechos de autor al utilizar imágenes de internet como botones, se ha preferido crearlos desde 0.

Las pantallas se han diseñado previamente en papel para situar de la manera más eficiente los botones (analizando la situación de la pantalla).

En la figura 3 y 4 se pueden ver las pantallas del modo museo y modo combate respectivamente. En el caso del modo combate, la pantalla finalmente se cambió para darle más visibilidad al usuario, para conseguir esto se

posicionaron los botones en la parte baja de la pantalla, en el apartado resultados se puede ver el resultado final.

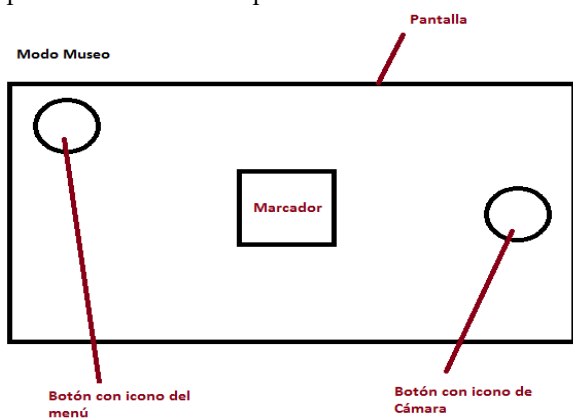


Figura 3: Pantalla modo museo

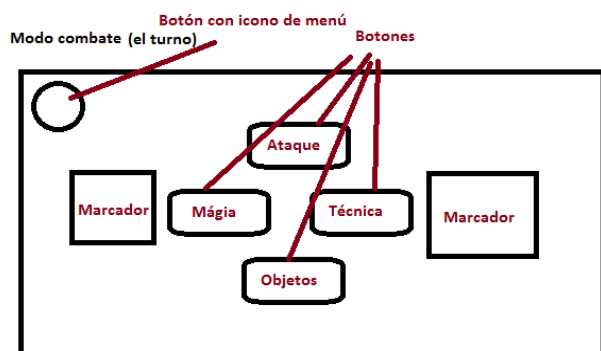


Figura 4: Pantalla modo combate

Para el análisis de recursos se ha utilizado una herramienta diferente a UnityProfiler (herramienta de Unity para hacer un análisis de los recursos utilizados). Esto ha sido debido a que UnityProfiler es de pago. En su lugar se ha utilizado una aplicación para móvil para analizar el consumo que tiene el videojuego. La aplicación que se ha utilizado es CPU Monitor.

CPU Monitor es una app para móvil que se puede encontrar en Google Play de forma gratuita. Además de poder analizar los recursos utilizados, tiene graficas representativas del voltaje y la temperatura del móvil. Permite hacer tests de 2 minutos en los que pones en ejecución el test y lo dejas en segundo plano.

Para hacer un análisis de cuanto aguantaría la batería se han detenido todas las aplicaciones posibles y desconectado de internet el móvil.

Se ha hecho un cálculo **aproximado** del consumo teniendo en cuenta el tiempo que tarda en bajar un 1% en la batería. Ha tardado aproximadamente 1 minuto y medio con el brillo de la pantalla al máximo, lo que significa que para el 100% tardará aproximadamente 130 minutos (2.17 horas).

En la figura 5 se puede ver que la carga de memoria es bastante elevada comparada con otras aplicaciones, algo que ya se esperaba ya que los gráficos 3D con la AR consumen bastantes más recursos que una aplicación de mensajería como puede ser "Whatsapp".

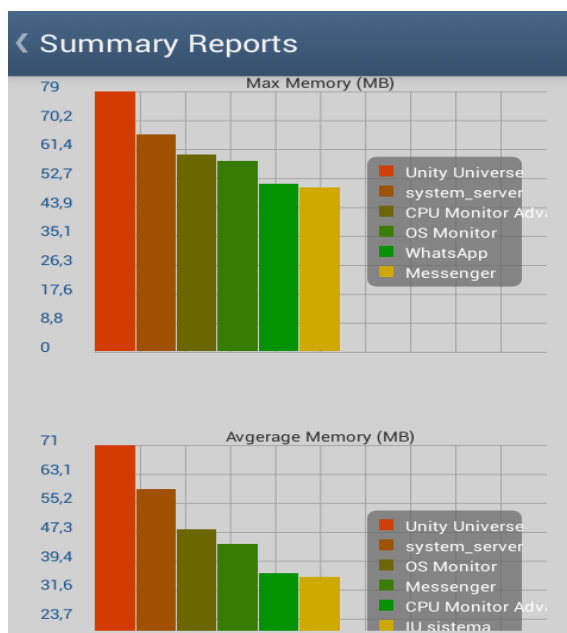


Figura 5: Comparativa memoria

4.5 Módulo de Lógica

Las tareas de este módulo están relacionadas con la parte de control del combate y la parte funcional (acciones de los botones) de las pantallas. También se han realizado diagramas para agilizar la posterior codificación.

Esta parte se ha desarrollado con el compilador MonoDevelop, que ya viene integrado con Unity, como lenguaje de programación se ha utilizado C#. Los diagramas se han realizado con Microsoft Visio 2013, un software diseñado especialmente para hacer diferentes tipos de diagramas.

Se ha implementado el menú principal, el cual tiene un *label* que actúa de título, un botón para ir al modo combate y otro para ir al modo museo. Además al hacer una carga invoca una pantalla de carga para dar un *feedback* al usuario y no pueda pensarse que se ha bloqueado la aplicación.

Se ha implementado un modo museo, el cual consta de 2 botones, uno para volver al menú y otro para hacer una foto. La foto generada se guarda en la carpeta raíz del juego. Este modo es un modo que sirve para poder observar de cerca a tus personajes y poder hacerles fotos.

Para la persistencia de datos en un primer momento se iba a utilizar como persistencia una base de datos SQLite, pero después de hacer un diagrama entidad relación se llegó a la conclusión que por el momento no era necesaria debido a su simpleza. En su lugar como persistencia de datos se han utilizado ficheros XML.

Se ha generado un diagrama de clases para facilitar la posterior codificación pensando qué clases existen, qué atributos tiene cada una de ellas y los métodos de cada entidad. Con este diagrama también se ha podido analizar cómo se comporta cada clase con el resto de clases.

El diagrama de clases se ha hecho conforme la nomenclatura básica de UML.

Este diagrama se puede ver en el anexo 1.

A continuación se explican los métodos principales de las clases del modo combate.

- **Clase combate:**

Es la clase que se encarga del “bucle” del juego, la clase central. A ella se relacionan la clase Personaje y por extensión la clase Ataque especial.

- **Start ():** método propio de Unity que se ejecuta nada más cargar la escena.

Inputs: Nada

Outputs: Nada

- **onGui ():** método propio de Unity que se encarga del refresco en pantalla de los botones y sus acciones, se ha utilizado básicamente para la GUI.

Inputs: Nada

Outputs: Nada

- **Update ():** método propio de Unity que se encarga del refresco de la lógica del programa, aquí hemos puesto la lógica como el control de vida.

Inputs: Nada

Outputs: Nada

- **CargaAtributos (Nombre):** método que sirve para cargar los atributos de un XML y cargarlos directamente en los personajes. Se le pasa el nombre del personaje para identificar los atributos del personaje que buscamos.

Inputs: nombre:string

Outputs: Nada

- **RandomEnemyAttack ():** método que elige el ataque que hará el enemigo de forma aleatoria al ser atacado y llama a la animación del ataque.

Inputs: Nada

Outputs: Nada

- **Clase Personaje:**

Clase que hace la función de personaje, tiene los atributos del personaje además del método que un personaje puede realizar, el de ataque.

- **Ataque (Personaje):** método que ejecuta el ataque del personaje. Le enviamos el personaje defensor y hace el cálculo del daño, le quita la VIT al enemigo y llama a la animación.

Inputs: Nada

Outputs: Nada

- **Clase Ataque Especial:**

Clase que hace la función de hechizo o habilidad, como tal posee los atributos que tienen estos ataques (nombre, coste, elemento y bonificador).

Se ha implementado un modo museo, el cual consiste en un combate entre el jugador y la máquina. En este modo hay 3 botones de ataque (Ataque, Hechizo y Habilidad) y un botón para volver al menú. Una vez tocas cualquiera de los botones de ataque, el ataque se realiza y en caso de que el enemigo no haya muerto contraataca.

En caso de que el enemigo muera se muestra una pantalla en la que puedes repetir el combate o salir al menú principal.

En el anexo 2 se puede observar el diagrama de flujo del modo combate, este diagrama se realizó para hacer más sencilla la codificación del combate. En este diagrama se puede ver toda la lógica del modo combate una vez el proyecto termine, para el TFG se decidió llegar solo a hacer la parte central (la rama del ataque normal) con el final de batalla y las opciones de fin de batalla.

Por último, se han realizado tests funcionales del videojuego para observar que funcione bien y poder saber si aparece algún error.

Los tests de usabilidad se han llevado a cabo por parte de un grupo de personas escogidas. Este test se ha realizado observando como interactuaban los usuarios y pasándoles una encuesta una vez terminada la demostración.

La siguiente tabla muestra los resultados de los tests funcionales realizados y su estado.

Prueba	Resultado que debería dar	Estado
En el modo combate, tocar el botón Ataque.	Hace la animación de ataque normal (estira un brazo)	Correcto
En el modo combate, tocar el botón Hechizo.	Hace la animación de ataque especial (estira los 2 brazos)	Correcto
En el modo combate, tocar el botón Habilidad.	Hace la animación de ataque especial (estira los 2 brazos)	Correcto
En el modo combate, tocar el botón para volver al menú.	Te lleva al menú.	Correcto
En el modo combate, tocar cualquier botón de ataque.	Desaparecen los botones durante las animaciones.	Correcto
En el modo museo, tocar el botón de la cámara.	Hace una foto y la guarda en la carpeta del proyecto	Correcto
En el modo museo, tocar el botón para volver al menú.	Te lleva al menú.	Correcto
En el menú principal, toca el botón de Modo museo.	Te lleva al modo museo	Correcto
En el menú principal, toca el botón de Combate libre.	Te lleva al modo combate libre	Correcto
Poner la aplicación en segundo plano y volver a abrirla.	La ejecución del videojuego se pausa y cuando vuelves a abrir la aplicación, sigue por donde se había quedado.	Correcto
En el modo combate atacar al enemigo hasta que sobrepase su vida	El enemigo cae y aparece una pantalla con la cual puedes salir o repetir el combate.	Correcto
En el modo combate atacar	El enemigo pierde la vida según el daño del ataque	Correcto

5 RESULTADOS

En este apartado se presentan los resultados finales que se han logrado en el transcurso del TFG con la consecución de las tareas explicadas en los apartados anteriores.

5.1 Módulo de Visualización

En la figura 6 se puede ver al personaje diseñado y en la figura 7 el personaje descargado. Se puede observar que el nuestro es claramente menos complejo, esto es debido a que se quería crear un personaje y sus animaciones para poder aprender un poco de cada fase de un videojuego, pero sin darle demasiada dedicación a esta parte.

El personaje creado (Froyo) se ha creado con una esfera que ha formado el cuerpo y los cuernos, un cono a modo de sombrero, 2 planos como telas y 2 cubos moldeados con la herramienta *extrude* para darle la forma deseada a los brazos.

Para hacer los cuernos simplemente se cogió uno de los vértices y se deformó hacia atrás, con la herramienta *mirror* lo hizo automáticamente en la otra mitad de la esfera. El cono se escaló en las dimensiones “x” y “z” para que cupiera en la esfera, ya que esta no es una esfera perfecta. Finalmente se ha puesto un esqueleto en ambos brazos (solo en los brazos) para hacer las animaciones.

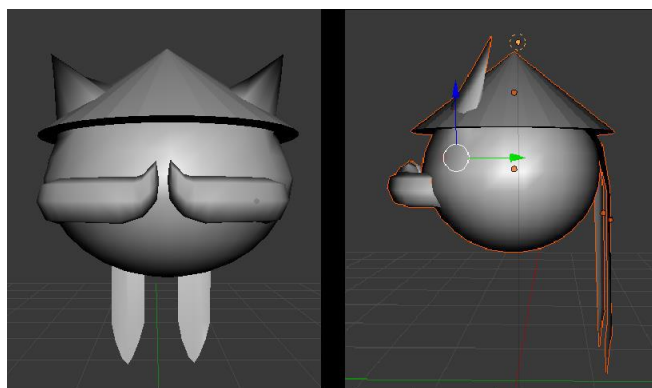


Figura 6: Froyo vista frontal y lateral

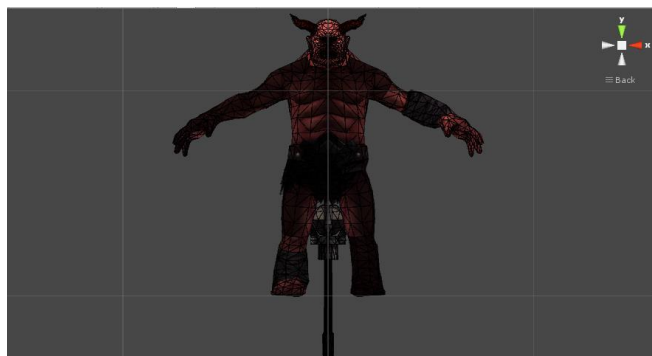


Figura 7: Personaje Cyclope

5.2 Módulo de GUI

Los botones diseñados se pueden observar en la figura 8. El botón de la derecha lleva al menú principal y está presente en el mismo lugar en todas las pantallas excepto en el menú principal. El botón de la izquierda hace una fotografía a la escena y la guarda en la carpeta del proyecto, solo está en el modo museo.

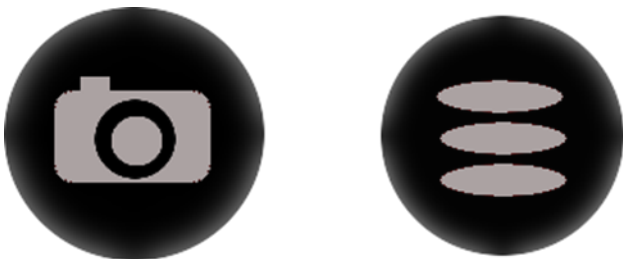


Figura 8: Boton de foto y botón de menú respectivamente

En los resultados del módulo de lógica se puede ver como ha quedado la pantalla de cada modo y compararlas con los diseños principales mostrados en el punto “4.4 Módulo de GUI”.

5.3 Módulo de Lógica

Al ejecutar la aplicación, lo primero que aparece es el menú principal. Este menú muestra el título del videojuego y te da la opción de ir al modo museo y al modo combate.

En las siguientes figuras se presenta como quedan finalmente cada uno de los modos. Como se puede ver en la figura 9 se ha respetado el diseño de pantalla realizado en papel, posicionando el botón de menú en la esquina superior izquierda y el botón para hacer una fotografía a la derecha.



Figura 9: Modo museo

En cambio como se puede ver en la figura 10, la pantalla del modo combate ha sido cambiada para darle más visibilidad al jugador, posicionando los botones de acción (Ataque, Hechizo y Habilidad) en la parte baja de la pantalla.

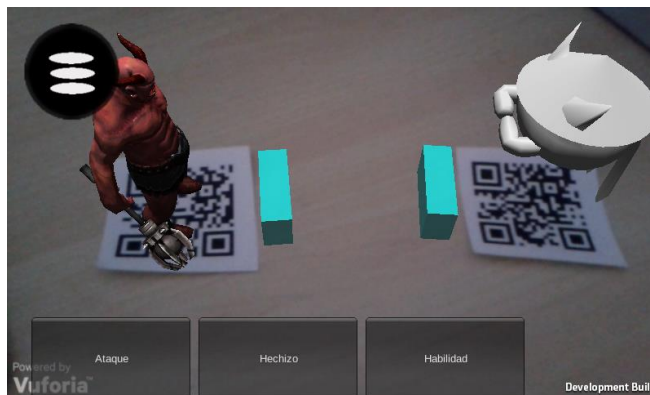


Figura 10: Modo combate

En el modo combate, al terminar el combate aparece la pantalla de fin de combate que te dice si has ganado o perdido y te da la opción de una revancha o de salir al menú principal.

Como se ha comentado en el apartado “4.5 Módulo de lógica” se ha decidido utilizar archivos XML como persistencia por su simpleza. En la figura 11 se puede ver el archivo de guardado de los personajes.

```
<Personaje>
  <Nombre>Froyo</Nombre>
  <Clan>Ninguno</Clan>
  <Elemento>Agua</Elemento>
  <VIT>200</VIT>
  <PM>50</PM>
  <PH>10</PH>
  <Velocidad>20</Velocidad>
  <ATQM>40</ATQM>
  <ATQF>20</ATQF>
  <DEFM>50</DEFM>
  <DEFF>30</DEFF>
  <Ataques>
    <Nombre>Hielo</Nombre>
    <Nombre>Frio del universo</Nombre>
  </Ataques>
</Personaje>
<Personaje>
  <Nombre>Cyclo</Nombre>
  <Clan>Ninguno</Clan>
  <Elemento>Tierra</Elemento>
  <VIT>3</VIT>
  <PM>0</PM>
  <PH>2</PH>
  <Velocidad>10</Velocidad>
```

Figura 11: XML de los personajes

6 CONCLUSIONES

Con este TFG he podido observar que desarrollar una aplicación para móvil tiene más dificultad en el aspecto de los recursos que hacerlo para ordenador, ya que este último tiene más potencial en todos los componentes y no tiene problemas de batería.

El software generado es bastante escalable, ya que se pueden añadir nuevos personajes y nuevos ataques sin retocar o retocando mínimamente la organización de clases. También se puede añadir un nuevo modo multijugador teniendo que crear la parte de la conexión entre móviles y una pantalla de selección de personaje, pero no habría que modificar las clases ya creadas.

Para el modelado y la animación de los personajes se ha utilizado Blender y Unity respectivamente. Estos softwares son bastante potentes, ya que te proporcionan muchas herramientas para poder diseñar videojuegos, animaciones y otro tipo de productos. En contramedida, al proporcionar tantas herramientas, si nunca has trabajado con el software puedes tener problemas por la complejidad de este. Después de mi experiencia con este proyecto, diría que si quieres utilizar estas herramientas para un proyecto y nunca las has utilizado, antes de comenzar a desarrollar con ellas tengas un tiempo de aprendizaje en el que poder aprender unos mínimos. Tanto la web oficial de Blender como la de Unity ofrecen documentación, manuales y tutoriales para aprender a utilizar sus herramientas. Una vez se ha realizado un pequeño aprendizaje su utilización no es difícil.

Para los marcadores la mejor solución que he encontrado es la de los códigos QR. Tienen una complejidad muy buena para la detección, logrando una rápida detección y buena robustez (pudiendo tapar gran parte del marcador y que el modelo siga viéndose). Para diseñar estos códigos se ha utilizado el siguiente generador de códigos QR: <http://www.codigos-qr.com/generador-de-codigos-qr/>

Para capturar los requisitos se hizo una encuesta mediante SurveyMonkey (<https://es.surveymonkey.com/>). Es una Web muy intuitiva en la que se pueden hacer encuestas y poder difundirlas mediante un link a las personas que quieras. Esta herramienta además de los resultados también te ofrece gráficas y números generales de la encuesta.

El AR se ha conseguido mediante la librería Vuforia. Esta librería es gratuita y se puede utilizar para Android, iOS y para Unity (a modo de extensión). En mi caso la he utilizado junto a Unity y no ha sido complicada de utilizar. En la propia Web hay un foro dónde puedes ayudarte si tienes dudas, además al instalar la extensión te estructura automáticamente todas las carpetas. Por otra parte los códigos propios del AR son editables, así tienes algo más de flexibilidad.

6.1 Líneas futuras

El seguimiento de este proyecto conllevaría la realización de las siguientes tareas:

- Hacer más personajes para tener un modo combate en el que se pudiera elegir entre un plantel más variado.
- Completar el modo combate. Para completarlo se cambiarían los turnos actuales por turnos basados en el tiempo, se añadirían todos los atributos del sistema de juego y se completarían las 2 ramas del diagrama de flujo que quedan por implementar (Hechizo y Habilidad).

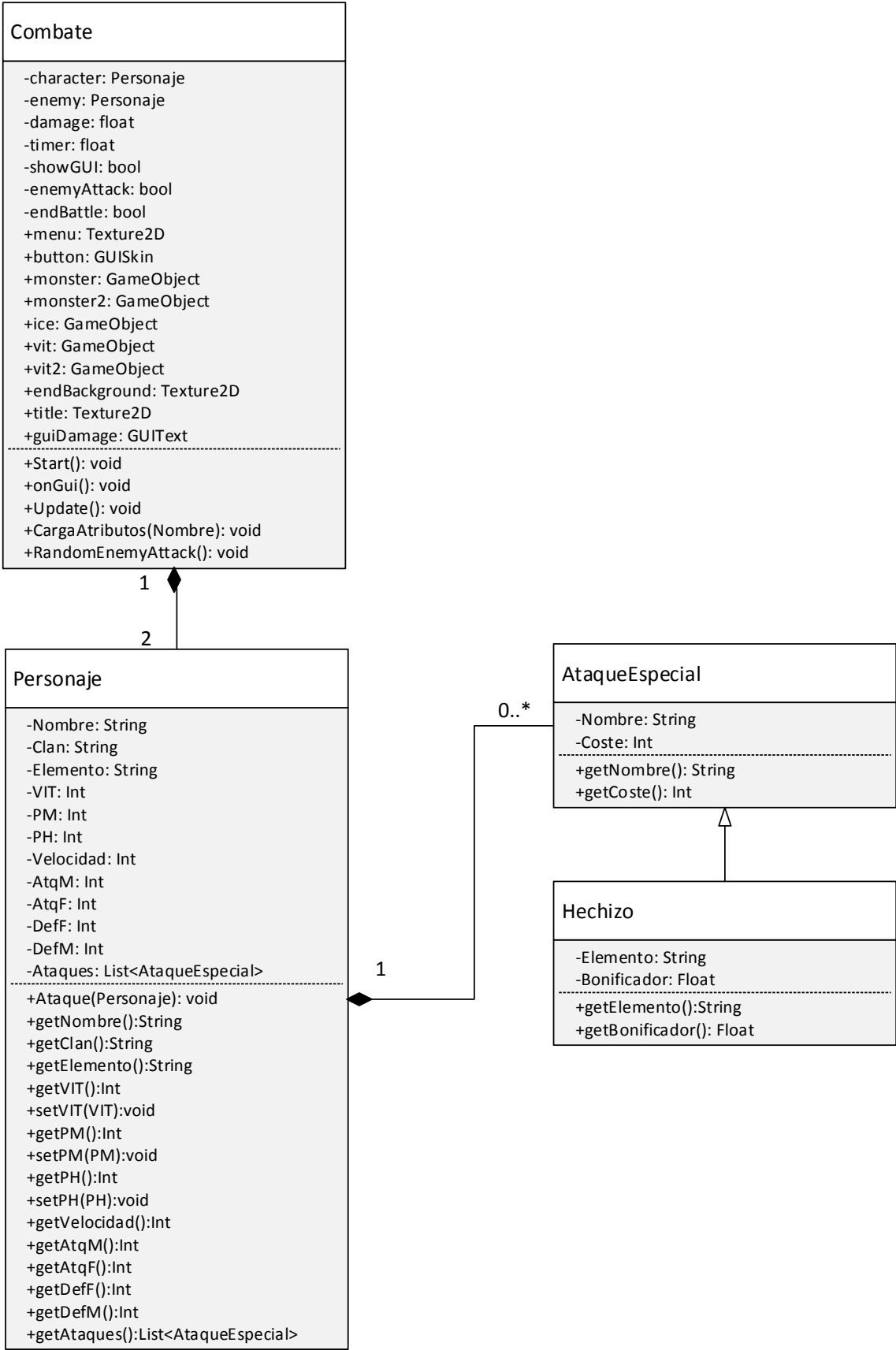
- Implementar un modo de combate entre 2 jugadores mediante la conexión móvil "bluetooth".
- Corregir algunos aspectos y bugs encontrados gracias al test de usabilidad.

BIBLIOGRAFIA

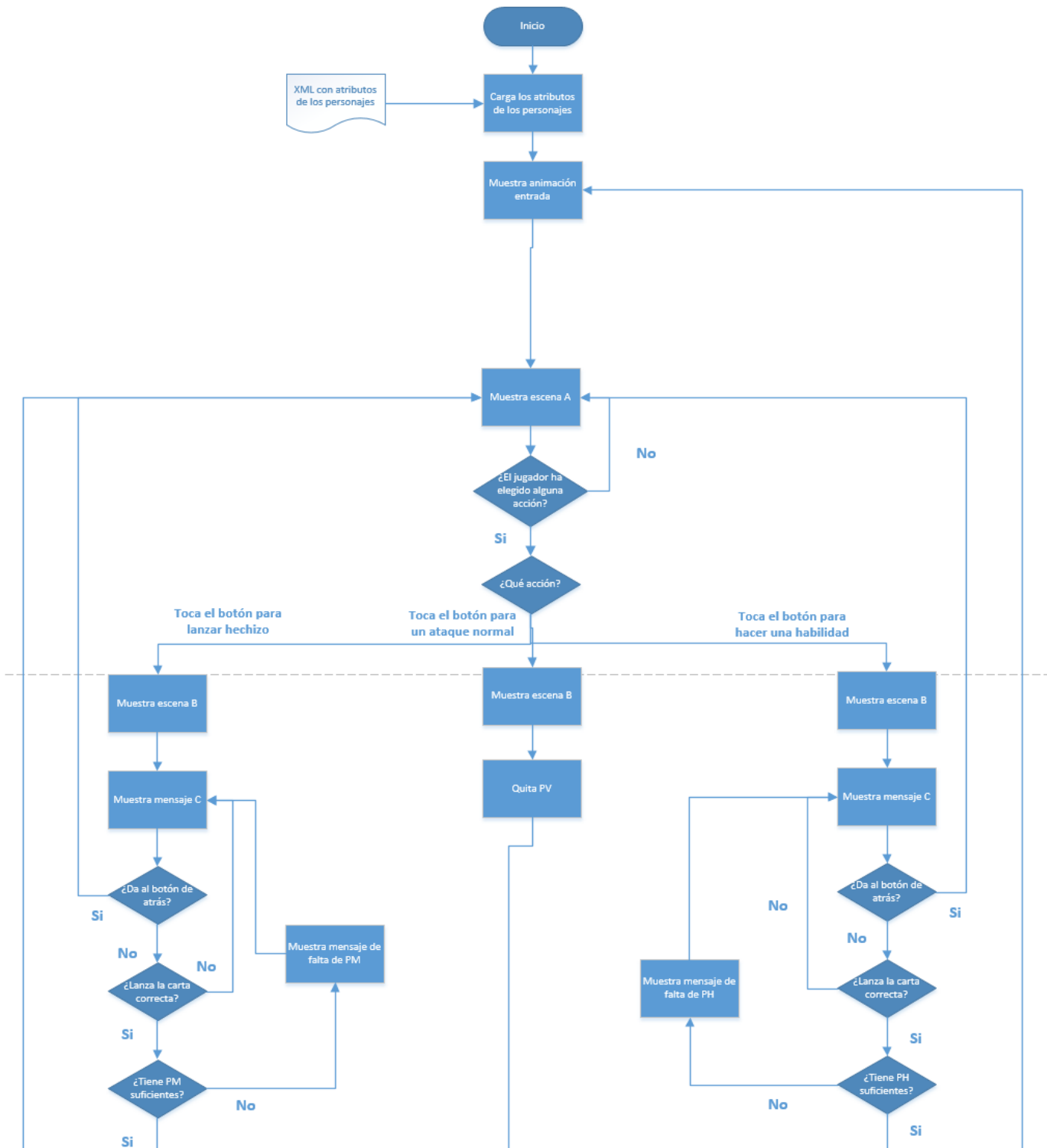
- [1] OCULUS, V. R. "Oculus Rift". Available from WWW:<<http://www.oculusvr.com/rift>>, 2015. [Consulta: 23 Junio 2015]
- [2] FELTHAM, Jamie. "Project Morpheus: The story so far", Mayo 2015, <<http://vrfocus.com/archives/15791/project-morpheus-the-story-so-far/>>. [Consulta: 23 Junio 2015]
- [3] Fundación Telefónica, "Realidad Aumentada: Una nueva lente para ver el mundo", Sainz, R. M. (2011), pp. 7-10.Referència 2 [Consulta: 6 Marzo 2015]
- [4] Muy Interesante, "Realidad aumentada, la última revolución digital", Javier Pedreira (12/07/2010) <<http://www.muyinteresante.es/tecnologia/articulo/realidad-aumentada-la-ultima-revolucion-digital>> [Consulta: 9 Marzo 2015]
- [5] Virtualama, "Realidad aumentada aplicada a la medicina"(18/07/2014), <<http://www.virtualama.com/blog/realidad-aumentada-y-medicina/>> [Consulta: 9 Marzo 2015]
- [6] Web PlayStation (Invizimals), Marzo 2015, <<http://es.playstation.com/psn/games/detail/item158726/Invizimals%E2%84%A2>>, [Consulta: 8 Marzo 2015]
- [7] Web PlayStation (Wonderbook: el libro de pociones), Marzo 2015 <<https://www.playstation.com/es-es/games/wonderbook-book-of-potions-ps3>> [Consulta: 8 Marzo 2015]
- [8] Web PlayStation (EyePet), Marzo 2015 <<https://www.playstation.com/es-es/games/eyepet-ps3>> [Consulta: 8 Marzo 2015]
- [9] Vuforia. Página principal, Marzo 2015 <<https://developer.vuforia.com/>>
- [10] Blender. Página principal, Marzo 2015 <<https://www.blender.org/>>
- [11] Unity. Página principal. Marzo 2015 <<https://unity3d.com/es>>
- [12] UNITY3D. Asset store <<https://www.assetstore.unity3d.com>> [Consulta: 8 Mayo 2015]
- [13] DENSO WAVE INCORPORATED. History of QR code. <<http://www.qrcode.com/en/history>> [Consulta: 18 Mayo 2015]

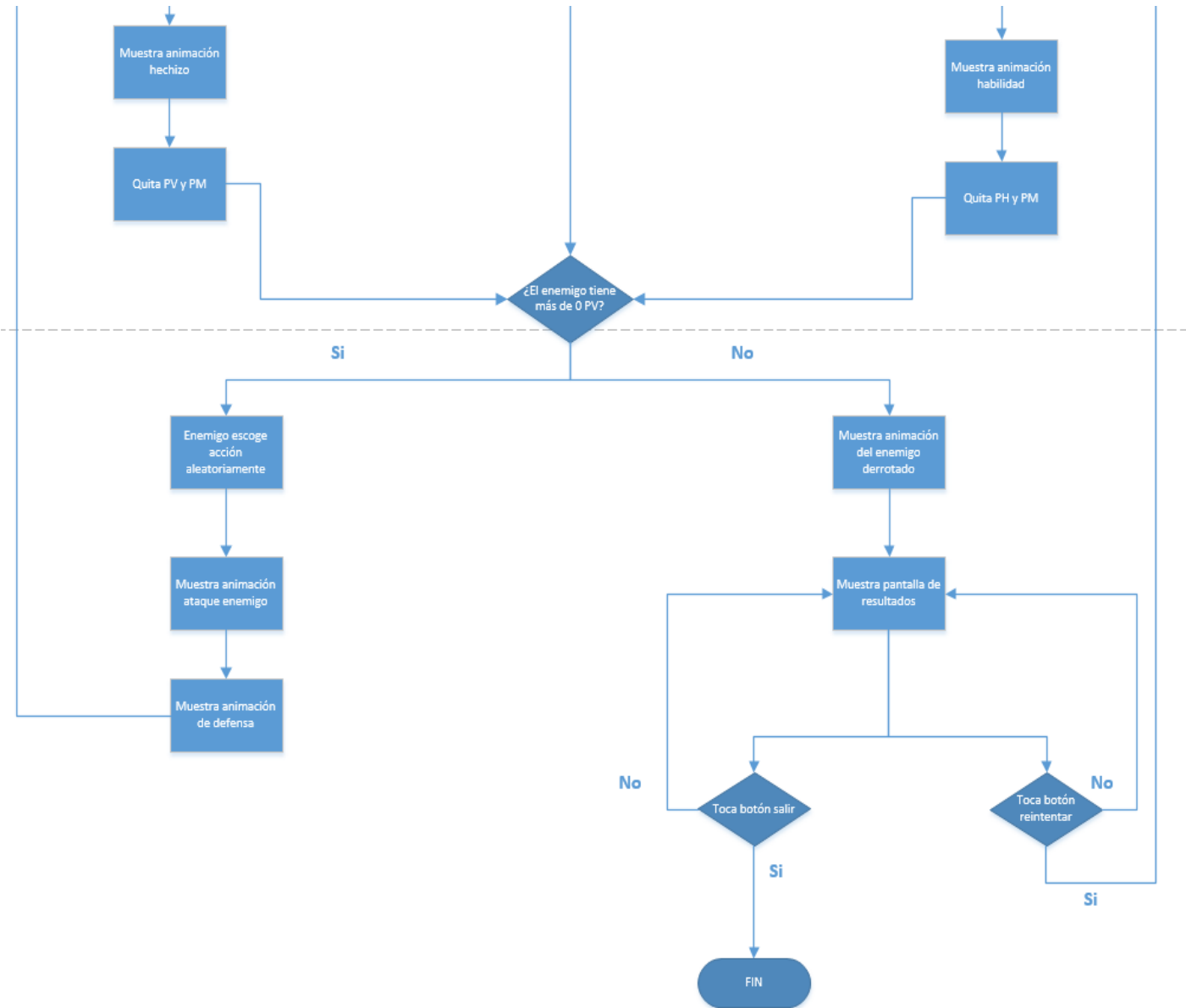
ANEXO

A1. Diagrama de clases



A2. Diagrama de flujo





Escena A: Significa la escena del combate sin los botones de comando.

Escena B: Significa la escena del combate con los botones de comando.

Mensaje C: Es un mensaje que se muestra para que el jugador lance la carta.